

RECEIVED
CENTRAL FAX CENTER

JAN 18 2007

CASE NO.: HSJ920030256US1
Serial No.: 10/756,123
January 17, 2007
Page 4PATENT
Filed: January 12, 2004Remarks

The Section 101 rejections of all currently pending claims (1-9) and Section 112 rejections of independent claim 1 are noted. Applicant will present arguments below as to the incorrectness of these rejections.

Claims 1-9 have been rejected under 35 U.S.C. §103 as being unpatentable over Drawswandter et al., USPP 2003/0210275 (hereinafter "ref (a)") in view of Broussard et al., USPP 2003/0159130 ("ref b").

The fact that Applicant has focussed its comments distinguishing the present claims from the applied references and countering certain rejections must not be construed as acquiescence in other portions of rejections not specifically addressed.

Rejections Under 35 U.S.C. §101

All pending claims have been rejected under 35 U.S.C. §101 for being directed to non-statutory subject matter. Underpinning all of the rejections at bottom appear to be (although the grammar of the rejections makes it difficult to be certain) bare unsupported allegations that graphical user interfaces are mere software and hence non-statutory since they do not represent useful, tangible, and concrete results. This is a somewhat interesting position for the Patent Office to take. A GUI is not mere music, literary work, or simple collection of data as contemplated by MPEP §2106.01. Anyone who operates a computer these days uses a GUI to do so. It is a tool without which almost no PC would be operable.

Moreover, the Patent Office has issued over 700 patents with "GUI" or "graphical user interface" in the title. Plainly, producing a tool such as a hammer must be considered to produce a useful, tangible, and

1189-25.AM1

CASE NO.: HSI920030256US1

Serial No.: 10/756,123

January 17, 2007

Page 5

PATENT

Filed: January 12, 2004

concrete result, and hammers are less ubiquitously used than GUIs. Not surprisingly, the allegation that producing a GUI is non-statutory is absolutely without a shred of legal authority: it is simply an unfounded and completely unexplained position taken by an examiner.

Nevertheless, in the interest in promoting prosecution Claim 1 now positively recites in its body what had been suggested previously in the preamble, namely, that the GUI page is used to configure a data pipeline. Configuring a useful data pipeline, a few example useful applications of which are discussed in the present background, cannot legitimately be said not to be a useful and tangible result.

Rejections Under 35 U.S.C. §112, First Paragraph

Claim 1 has been rejected under 35 U.S.C. §112, first paragraph based on the allegation that the "computer memory" lacks adequate written description, despite the fact that page 7, lines 10 and 11 of the specification explicitly state that "The entire pipeline system 10 can be implemented *in the memory of a computer.*"

MPEP §§2163.02 and 2163.04 inform in relevant part the written description inquiry. The objective test, according to the MPEP, is whether the description - here, a statement that the entire pipeline system 10 disclosed in the specification can be implemented in the memory of a computer - conveys to the skilled artisan that the inventor possessed the "computer memory" of Claim 1. Such a written description is presumed to be adequate, subsection (04), and unless *a preponderance of the evidence* can be mustered by the examiner to show otherwise. Here, the rejections come unaccompanied by any explanation, much less any showing resembling a "preponderance of the evidence", why the skilled artisan, having been told that the system of the

1189-25.AM1

CASE NO.: HSI920030256US1

Serial No.: 10/756,123

January 17, 2007

Page 6

PATENT

Filed: January 12, 2004

present invention can be embodied in a computer memory, would have failed to grasp that the inventor possessed the "computer memory" of Claim 1. The rejection fails to satisfy the MPEP and thus fails.

Rejections Under 35 U.S.C. §112, Second Paragraph

Claim 1 has been rejected under 35 U.S.C. §112, second paragraph based on the bare unexplained allegation that the "computer memory" of Claim 1 is "unclear". However, "breadth is not indefiniteness", MPEP §2173.04, and while the term "computer memory" certainly encompasses a large number of devices, the skilled artisan is adequately informed as to the scope of the claim, i.e., that it encompasses any computer memory capable of meeting the limitations of Claim 1.

Rejections Under 35 U.S.C. §103

Independent Claim 1

Claims 1-9 have been rejected under 35 U.S.C. §103 as being unpatentable over ref (a) in view of ref b. Ref (a) is directed to compiling code, not to processing a pipeline of data for the various purposes described in the present application. With this in mind, the rejection alleges that ref (a), paragraph 23 teaches that the main GUI window is a pipe input set window configured to permit a user to define a type of pipe input set data, and that a GUI page based on the type is configurable, but Applicant respectfully does not discern the alleged teaching in paragraph 23. Instead, paragraph 23 discloses that the same interface module can be used to describe compiler commands for different command line interface (CLI) compiler applications.

1189-25.AM1

CASE NO.: HSI920030256US1
Serial No.: 10/756,123
January 17, 2007
Page 7

PATENT
Filed: January 12, 2004

A command is not a type of pipe input set data as now recited in Claim 19. The commands evidently can be used to specify the name of the file to be compiled, but, like a compiler command, a file name is not a data type. Applicant notes that the examiner has not provided a claim construction for "data type" to aid Applicant in understanding the basis of the rejection (e.g., is it the examiner's position that "data type" encompasses compiler commands?), but in any case Applicant notes that claim terms must be construed as one skilled in the art would construe them, MPEP §2111.01, and no evidence has been pointed to in the references that skilled artisans regard compiler commands or file names to be "data types".

With respect to the obviousness rejection of Claim 1 based on combining refs (a) and (b), several observations are germane. First, the rejection alleges that ref (a), paragraphs 17 (lines 1-5) and 20 teach a pipe input set window configured to permit a user to define a type of pipe input set data, but as mentioned above, the point of ref (a) appears to be to provide a GUI that supports different CLI commands, not different input data *types*. Indeed, this is precisely what is taught in the relied-upon portion of paragraph 17 of ref (a), while paragraph 20 teaches that the file to be compiled - not an input data type - can be specified. Since a file to be compiled can contain data of an indeterminant and, in ref (a), unspecified type, merely specifying the file name is not specifying a data type.

Applicant also observes that the proposed combination of ref (b), used as a teaching of Java reflection to arrive at the claimed limitation of using Java reflection to generate an instance of the input data type class, appears to fail to comply with the requirements of MPEP §2143.01, in that the prior art motivation to combine is missing giving the following facts. While ref (b), paragraph 25 teaches that Java reflection can be used to generate GUIs in what appears to be an object-oriented programming (OOP) environment, ref (a) does not

1189-25.AM1

CASE NO.: HSI920030256US1
Serial No.: 10/756,123
January 17, 2007
Page 8

PATENT
Filed: January 12, 2004

appear to be such an environment. Specifically, recall that ref (a) is directed to compilers, and that there has been no evidence identified that the use of Java, much less Java reflection, is an appropriate modification of a compiler application. In effect, the two references are apples and oranges, and only through hindsight of knowing what the present claims recite would it appear to be suggestive to combine them.

Furthermore, even if the references were combined as proposed, Claim 1 would not result. Specifically, it does not appear in the relied-upon sections of ref (b) that Java reflection is used to undertake the specific task recited in Claim 1, namely, using Java reflection to generate an instance of a class that itself has been derived by translating an input data type using a configuration file. Applicant is unable to discern any teaching in the references to translate an input data type into a class using a configuration file, much less to then use Java reflection to generate an instance of this specifically defined class.

It appears that an attempt has been made to respond to some, but certainly not to all, of the above observations in paragraph (e), pages 25 and 26 of the Office Action. Because of the grammar it is difficult for Applicant to know exactly what the gravamen of the response is, but it appears to Applicant that the response is based on a hypothetical equivalence of Java to other programming languages and then to a leap to discern from the hypothetical equivalence some inexplicable suggestion to use Java reflection. The glaring point remains, however, that Claim 1, which is directed to a GUI for configuring a data pipeline, stands rejected on a primary reference that has nothing to do with data pipeline GUIs but rather is related to compilers.

Dependent Claim 2

1189-23.AM1

CASE NO.: HSI920030256US1
Serial No.: 10/756,123
January 17, 2007
Page 9

PATENT
Filed: January 12, 2004

In addition to the above, the allegation that paragraph 23 of ref (a) teaches a pipe input set window and a GUI page that require no programming apart from an initial core code is wrong. Paragraph 23 nowhere mentions "GUI" or any cognizable synonym to "pipe input set window" and indeed appears to teach exactly the opposite of "requiring no programming apart from an initial core code". Consider: "if a different external tool CLI application is to be accessed, *the only change that needs to be done to interface 10* is to provide a new data structure...." Thus paragraph 23 explicitly teaches that the relied-upon module 10 must change, militating toward patentability of Claim 2.

Dependent Claim 3

In addition to the above, the allegation that paragraph 37 of ref (a) teaches an incremental GUI wherein GUI pages for new pipe components can be added incrementally without changing existing code is without merit. Paragraph 37 teaches that a GUI generator module 22 does not change depending on something called a "CLI application" but that what the GUI shows does change. This is simply not a cognizable teaching of the incremental addition of GUI pages for new pipe components.

Dependent Claim 4

In addition to the above, the allegation that paragraph 35 of ref (a) teaches that a new pipe module is based on a pre-existing module type is without merit. There is nothing in paragraph 35 about pipe modules as that term is understood by the skilled artisan or about module types. Instead, paragraph 35 teaches that a GUI that is specific to a command verb (of a CLI application, not a data pipeline) is generated.

118925.AM1

CASE NO.: HSI920030256US1

Serial No.: 10/756,123

January 17, 2007

Page 10

PATENT

Filed: January 12, 2004

Dependent Claim 5

In addition to the above, the allegation that paragraph 39 of ref (a) teaches Claim 5 is without merit. Paragraph 39 appears to be directed to command lines being compiled, something that would appear to have nothing to do with data pipelines.

Dependent Claim 6

In addition to the above, the allegation that paragraphs 17 and 20 of ref (a) teaches Claim 6 is without merit. Claim 6 requires a GUI to define a set of interfaces with each interface including plural functions. It also requires that the GUI include a GUI representation part and a storage part, with the GUI representation part defining how something is displayed and the storage part defining how GUI parameters are stored in an external storage. In contrast, the relied-upon first five lines of paragraph 17 has nothing to do with a modular GUI but instead simply states that a GUI described a "command line". Nothing about interfaces, functions, or GUI parts describing display and storage. Likewise, paragraph 20 discussed command lines that can be used to correct floating point division errors and the name of the file to be compiled. Paragraph 20 indeed discloses that descriptions for groups of command line options can be stored in a data structure and associated with specific command verbs, but there is nothing here about interfaces, functions, or *GUI parts* describing display and storage.

Dependent Claim 7

In addition to the above, the allegation that paragraph 31 of ref (a) teaches Claim 7 is without merit. Claim 7 requires a *Pipe Output Set* tab for defining *PipeOutputSet* representative of a type of output data from

1189-25.AM1

CASE NO.: HSI920030256US1

Serial No.: 10/756,123

January 17, 2007

Page 11

PATENT

Filed: January 12, 2004

the pipeline, whereas paragraph 31 simply teaches that a command verb can be used to obtain a desired type of output, presumably during compilation, again a process having nothing to do with data pipelines. No tab is ever mentioned in this part of ref (a).

Dependent Claim 8

In addition to the above, the allegation that paragraphs 25 and 36 of ref (a) teaches Claim 8 is without merit. Claim 8 requires a *Storage For TupleSets* tab for defining an arbitrary number of elements contained in a *StorageForTupleSets* component of the pipeline, with individual input and output sets being definable for each element in the component. The relied-upon portions of ref (a) are directed only to the fact that two different GUIs are used for two different compilers (paragraph 25) and that command lines selected using the GUI are constructed according to the particular CLI application (paragraph 36). Nothing here about defining an arbitrary number of elements, much less one contained in any particular component of a data pipeline, and nothing about individual input and output sets being definable for each element in the component.

Dependent Claim 9


In addition to the above, the allegation that paragraph 35 of ref (a) teaches Claim 9 is without merit. Claim 9 requires a tab for defining an arbitrary number of *PipeModules* of the pipeline, with a type being selected for each *PipeModule* using the tab, the type defining at least in part the GUI. In contrast, paragraph 35 teaches generating a GUI specific to a command verb for software compiling, and has nothing to do with defining an arbitrary number of anything, much less data pipe modules, much less using a tab.

1189-25-AM1

CASE NO.: HSI920030256US1
Serial No.: 10/756,123
January 17, 2007
Page 12

PATENT
Filed: January 12, 2004

Respectfully submitted,



John I. Rogitz
Registration No. 33,549
Attorney of Record
750 B Street, Suite 3120
San Diego, CA 92101
Telephone: (619) 338-8075

JLR:jg

1189-25-AM1